

 Web Scarping

FLIPKART DATA ANALYSIS

Customer Sentimental
Analysis Using Python



BY AMANJOT SINGH

Customer Sentimental Analysis - Iphone 15 128gb

Objective:

As a Data Analyst at Flipkart, you have been tasked with gauging customer sentiment towards the iPhone 15 128GB model. The primary goal of this project is to analyze public perception and evaluate customer reactions by performing sentiment analysis on product reviews posted by users. By extracting and processing customer reviews, you will derive insights about the overall sentiment (positive or negative) surrounding the product, which can be useful for decision-making, improving customer experience, and identifying key areas for product improvement.

1. Data Collection (Web Scraping)

Tools: Selenium, BeautifulSoup

Task: Scrape 300+ customer reviews, including:

- **Username:** Reviewer's name
- **Rating:** 1 to 5 stars
- **Review Text:** Customer's feedback

```
#Importing required libraries
import requests
import time
import pandas as pd
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys

# Create empty lists to store the user data such as Name, City, Date of Purchase, Review & Rating
Names = []
Cities = []
Dates = []
Reviews = []
Ratings = []

# Assign the url of the flipkart website and use selenium to scrape data
url = "https://www.flipkart.com/apple-iphone-15-blue-128-gb/product-reviews/itmbf14ef54f645d?pid=MOBGTAGPAQNVFZZY&lid=LSTM0BGTAGPAQNVFZZYQRLPCQ&marketplace=FLIPKART"
```

```

driver = webdriver.Chrome()
driver.get(url)

while len(Names) < 320:

    time.sleep(2)
    soup = BeautifulSoup(driver.page_source, "html.parser")

    # Extract names
    names_elements= soup.find_all("p", {"class": "_2NsDsF AwS1CA"})
    for name in names_elements:
        Names.append(name.text)

    # Extract cities
    city_elements = soup.find_all("p", {"class": "MztJPv"})
    for city in city_elements:
        Cities.append(city.text)

    # Extract dates
    dates_elements = soup.find_all("p", {"class": "_2NsDsF"})
    for date in dates_elements:
        Dates.append(date.text)
    Actual_Dates = Dates[1::2]

    # Extract reviews
    reviews_elements = soup.find_all("div", {"class": "ZmyHeo"})
    for review in reviews_elements:
        Reviews.append(review.text)

    # Extract ratings
    ratings_elements = soup.find_all("div", class_ = "XQDdHH Ga3i8K")
    for ratings in ratings_elements:
        Ratings.append(ratings.text)

    # Try to click the "Next" button
    try:
        next_button = driver.find_element(By.XPATH,
"//span[text()='Next']")
        next_button.click()
        time.sleep(5)
    except:
        break

# Combine data into a DataFrame
df = pd.DataFrame({
    "Name": Names[:-1],
    "City": Cities[:-1],
    "Date": Actual_Dates[:-1],
    "Review": Reviews[:-1],

```

```
    "Ratings": Ratings[:-1]
})
```

2. Data Cleaning & Preprocessing

Tool: Pandas

Steps:

- Remove duplicates.
- Handle missing values.
- Convert text to lowercase.
- Remove special characters & extra spaces.
- Tokenize & remove stopwords.
- Apply lemmatization.

```
# Check the basic info of the dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 319 entries, 0 to 318
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   Name        319 non-null    object
 1   City        319 non-null    object
 2   Date        319 non-null    object
 3   Review      319 non-null    object
 4   Ratings     319 non-null    object
dtypes: object(5)
memory usage: 12.6+ KB
```

```
# Drop the duplicates from the dataframe
df1 = df.copy()
df1 = df1.drop_duplicates()
df1
```

		Name		City
Date \				
0	Mousam	Guha Roy	Certified Buyer, Matialihat	Oct, 2023
1		Ajin V	Certified Buyer, Balaghat	Oct, 2023
2		bijaya mohanty	Certified Buyer, Baleshwar	8 months ago
3		Prithivi Boruah	Certified Buyer, Bokajan	Oct, 2023
4		Nikhil Kumar	Certified Buyer, Meerut Division	Jan, 2024
..	

```

.
314 Suraj Kumar Gp Certified Buyer, Ranchi 11 months
ago
315 Sahil Khan Certified Buyer, Boisar 11 months
ago
316 kunal aggarwal Certified Buyer, New Delhi 11 months
ago
317 Suresh Babu Certified Buyer, Chennai 11 months
ago
318 Flipkart Customer Certified Buyer, Etah District 11 months
ago

```

```

                                Review Ratings
0                                Very niceREAD MORE      4
1                                High quality camera😊READ MORE      5
2 Just go for it.Amazing one.Beautiful camera wi...      5
3 Camera Quality Is Improved Loving ItREAD MORE      5
4 Switch from OnePlus to iPhone I am stunned wit...      5
..                                ...      ...
314                                AmazingREAD MORE      5
315 It can be more better if display refresh rate ...      5
316 Honest review after using it for a month! Best...      5
317 Great product again from apple.Worthy upgrade ...      5
318 Go for without hesitationREAD MORE      5

```

[297 rows x 5 columns]

Convert the Name column data into Title Case

```

df1['Name'] = df1['Name'].str.title()
df1.head()

```

```

      Name                                City      Date \
0 Mousam Guha Roy Certified Buyer, Matialihat Oct, 2023
1      Ajin V Certified Buyer, Balaghat Oct, 2023
2 Bijaya Mohanty Certified Buyer, Baleshwar 8 months ago
3 Prithivi Boruah Certified Buyer, Bokajan Oct, 2023
4 Nikhil Kumar Certified Buyer, Meerut Division Jan, 2024

```

```

                                Review Ratings
0                                Very niceREAD MORE      4
1                                High quality camera😊READ MORE      5
2 Just go for it.Amazing one.Beautiful camera wi...      5
3 Camera Quality Is Improved Loving ItREAD MORE      5
4 Switch from OnePlus to iPhone I am stunned wit...      5

```

Clean data of City column by removing unwanted characters/ part of string

```

df1['City'] = df1['City'].str.replace("Certified Buyer, ", "",
regex=False).str.strip()
df1.head()

```

	Name	City	Date	\
0	Mousam Guha Roy	Matialihat	Oct, 2023	
1	Ajin V	Balaghat	Oct, 2023	
2	Bijaya Mohanty	Baleshwar	8 months ago	
3	Prithivi Boruah	Bokajan	Oct, 2023	
4	Nikhil Kumar	Meerut Division	Jan, 2024	

	Review	Ratings
0	Very nice	4
1	High quality camera😊	5
2	Just go for it.Amazing one.Beautiful camera wi...	5
3	Camera Quality Is Improved Loving It	5
4	Switch from OnePlus to iPhone I am stunned wit...	5

Clean data of Review column by removing unwanted characters/ part of string and converting to lowercase

```
df1['Review'] = df1['Review'].str.lower().str.replace("read more", "",
regex=False)
df1.head()
```

	Name	City	Date	\
0	Mousam Guha Roy	Matialihat	Oct, 2023	
1	Ajin V	Balaghat	Oct, 2023	
2	Bijaya Mohanty	Baleshwar	8 months ago	
3	Prithivi Boruah	Bokajan	Oct, 2023	
4	Nikhil Kumar	Meerut Division	Jan, 2024	

	Review	Ratings
0	very nice	4
1	high quality camera😊	5
2	just go for it.amazing one.beautiful camera wi...	5
3	camera quality is improved loving it	5
4	switch from oneplus to iphone i am stunned wit...	5

#Convert Average_Polarity and Ratings to numeric to avoid plotting errors

```
df1["Average_Polarity"] = pd.to_numeric(df1["Average_Polarity"],
errors="coerce")
df1["Ratings"] = pd.to_numeric(df1["Ratings"], errors="coerce")
```

3. Sentiment Analysis:

- **Tool:** TextBlob
- **Steps:**
 - Analyze sentiment using TextBlob's polarity score (-1 to +1).
 - Classify sentiment:
 - Positive: Polarity ≥ 0.1
 - Negative: Polarity < 0.1
 - Store sentiment classification in the dataset.

```
# Import libraries for Sentimental analysis of review sentences
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import sent_tokenize
from nltk.tokenize import word_tokenize
from textblob import TextBlob
import string
```

```
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package stopwords to C:\Users\Amanjot
[nltk_data]   Singh\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to C:\Users\Amanjot
[nltk_data]   Singh\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to C:\Users\Amanjot
[nltk_data]   Singh\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

```
True
```

```
# Create a column called Reviews_t that stores tokenized sentences
from the Review column using the sent_tokenize function.
```

```
df1["Reviews_t"] = df1['Review'].apply(sent_tokenize)
df1.head()
```

	Name	City	Date	\
0	Mousam Guha Roy	Matialihat	Oct, 2023	
1	Ajin V	Balaghat	Oct, 2023	
2	Bijaya Mohanty	Baleshwar	8 months ago	
3	Prithivi Boruah	Bokajan	Oct, 2023	
4	Nikhil Kumar	Meerut Division	Jan, 2024	

	Review	Ratings	\
0	very nice	4	
1	high quality camera😊	5	
2	just go for it.amazing one.beautiful camera wi...	5	
3	camera quality is improved loving it	5	
4	switch from oneplus to iphone i am stunned wit...	5	

	Reviews_t
0	[very nice]
1	[high quality camera😊]
2	[just go for it.amazing one.beautiful camera w...
3	[camera quality is improved loving it]
4	[switch from oneplus to iphone i am stunned wi...

```

# Import mean from statistics for basic statistics
from statistics import mean

# Function created for assigning Polarity to the Reviews_t column
def get_polarity(sentences):
    return [TextBlob(sentence).sentiment.polarity for sentence in
sentences]

# Calls get_polarity function on the Reviews_t column to assign
polarity
df1['Polarity'] = df1['Reviews_t'].apply(get_polarity)

# Function created to calculate the average polarity of each review
(Average of polarity for each sentences in a review)
def calculate_average_polarity(polarities):
    return mean(polarities) if polarities else 0

# Calls calculate_average_polarity function on the Polarity column to
assign the average polarity for each review
df1['Average_Polarity'] =
df1['Polarity'].apply(calculate_average_polarity)
df1['Average_Polarity'] = df1['Average_Polarity'].round(2)
df1.head(10)

```

	Name	City	Date \
0	Mousam Guha Roy	Matialihat	Oct, 2023
1	Ajin V	Balaghat	Oct, 2023
2	Bijaya Mohanty	Baleshwar	8 months ago
3	Prithivi Boruah	Bokajan	Oct, 2023
4	Nikhil Kumar	Meerut Division	Jan, 2024
5	Akshay Meena	Jaipur	Nov, 2023
6	Flipkart Customer	Aizawl	Jan, 2024
7	Sheetla Prasad Maurya	Sultanpur	Oct, 2023
8	Raj Singh	Kolkata	Dec, 2023
9	Arunji Govindaraju	Chennai	Feb, 2024

	Review Ratings \
0	very nice 4
1	high quality camera😊 5
2	just go for it.amazing one.beautiful camera wi... 5
3	camera quality is improved loving it 5
4	switch from oneplus to iphone i am stunned wit... 5
5	so beautiful, so elegant, just a vowww😊♥ 5
6	awesome photography experience. battery backup... 5
7	best mobile phonecamera quality is very nice b... 4
8	for me its 10 out of 10👍 5
9	awesome product very happy to hold this. bette... 5

	Reviews_t \
0	[very nice]


```

1 [high quality camera😊]
2 [just go for it.amazing one.beautiful camera w...
3 [camera quality is improved loving it]
4 [switch from oneplus to iphone i am stunned wi...
5 [so beautiful, so elegant, just a vowww😊♥]
6 [awesome photography experience., battery back...
7 [best mobile phonecamera quality is very nice ...
8 [for me its 10 out of 10👍]
9 [awesome product very happy to hold this., bet...

```

	Polarity	Average_Polarity
0	[0.78]	0.78
1	[0.16]	0.16
2	[0.26666666666666666]	0.27
3	[0.6]	0.60
4	[0.0, 1.0]	0.50
5	[0.675]	0.68
6	[1.0, 0.7, 0.5]	0.73
7	[0.738]	0.74
8	[0.0]	0.00
9	[1.0, 0.5, 0.45555555555555555]	0.65

Function to assign the Class to the Polarity

```

def sentiment_class(polarity):
    if polarity > 0.75:
        return 'extremely positive'
    elif 0 < polarity <= 0.75:
        return 'positive'
    elif polarity == 0:
        return 'neutral'
    elif -0.75 <= polarity < 0:
        return 'negative'
    else:
        return 'extremely negative'

```

Calls sentiment_class function on the Average_Polarit column to assign the sentiment class

```

df1['Sentiment_Class'] =
df1['Average_Polarity'].apply(sentiment_class)
df1.head()

```

	Name	City	Date \
0	Mousam Guha Roy	Matialihat	Oct, 2023
1	Ajin V	Balaghat	Oct, 2023
2	Bijaya Mohanty	Baleshwar	8 months ago
3	Prithivi Boruah	Bokajan	Oct, 2023
4	Nikhil Kumar	Meerut Division	Jan, 2024

	Review Ratings \
0	very nice 4

	Reviews_t	
1	high quality camera😊	5
2	just go for it.amazing one.beautiful camera wi...	5
3	camera quality is improved loving it	5
4	switch from oneplus to iphone i am stunned wit...	5

Polarity \	Reviews_t	
0	[very nice]	
[0.78]		
1	[high quality camera😊]	
[0.16]		
2	[just go for it.amazing one.beautiful camera w...	
[0.26666666666666666]		
3	[camera quality is improved loving it]	
[0.6]		
4	[switch from oneplus to iphone i am stunned wi...	[0.0, 1.0]

	Average_Polarity	Sentiment_Class
0	0.78	extremely positive
1	0.16	positive
2	0.27	positive
3	0.60	positive
4	0.50	positive


```
# Calculates and prints the overall average polarity score of the
entire dataset of reviews
polarity_score = df1['Average_Polarity'].mean().round(2)
print(f'Average Polarity Score : {polarity_score}')
if polarity_score > 0.75:
    print('The Average Polarity Score is Extremely Positive')
elif 0 < polarity_score <= 0.75:
    print('The Average Polarity Score is Positive')
elif polarity_score == 0:
    print('The Average Polarity Score is Neutral')
elif -0.75 <= polarity_score < 0:
    print('The Average Polarity Score is Negative')
else:
    print('The Average Polarity Score is Extremely Negative')
```

Average Polarity Score : 0.51
The Average Polarity Score is Positive

4. Data Analysis & Insights

Tools: Pandas, Matplotlib, Seaborn

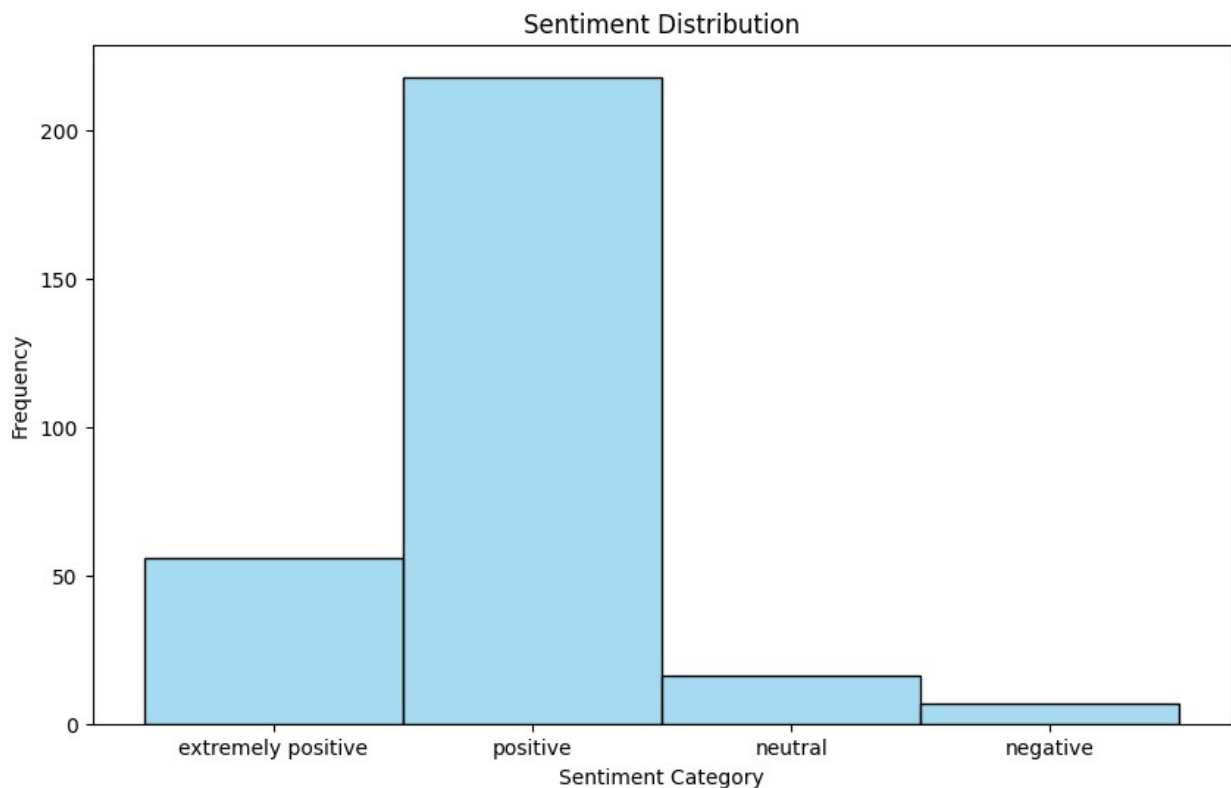
Steps:

- **Sentiment Distribution:** Count positive vs. negative reviews.
- **Rating vs Sentiment:** Check correlation between ratings & sentiment.

- **Word Cloud:** Identify frequent words in reviews.
- **Review Length Analysis:** Compare sentiment with review length.

```
# Importing libraries for visualisation
import matplotlib.pyplot as plt
import seaborn as sns

# Plots figure for Sentiment Distribution based on Sentiment Category
plt.figure(figsize=(10, 6))
sns.histplot(x=df1.Sentiment_Class, color='skyblue')
plt.title('Sentiment Distribution')
plt.xlabel('Sentiment Category')
plt.ylabel('Frequency')
plt.xticks(rotation=0)
plt.show()
```

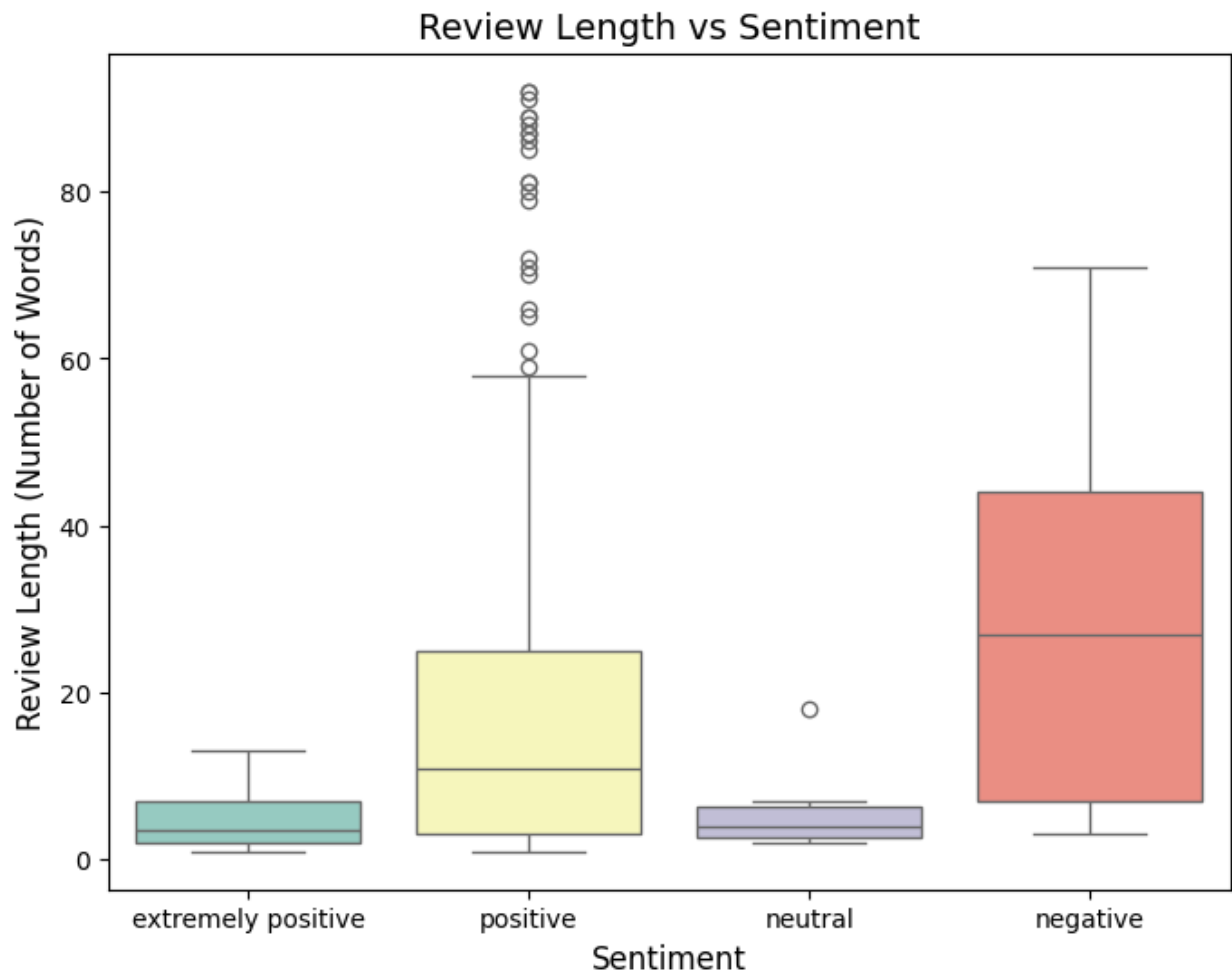


Sentiment Distribution

The sentiment analysis shows a dominance of positive reviews, with over 200 instances falling into this category. Extremely positive sentiments appear but at a lower frequency. Neutral reviews are less common, while negative feedback is the least represented. The data highlights a strong positive inclination, with very few users expressing dissatisfaction.

```
# Box Plot for Review Length by Sentiment
plt.figure(figsize=(8, 6))
```

```
sns.boxplot(x='Sentiment_Class', y='Review_Length', data=df1, hue =
'Sentiment_Class', palette='Set3')
plt.title('Review Length vs Sentiment', fontsize=14)
plt.xlabel('Sentiment', fontsize=12)
plt.ylabel('Review Length (Number of Words)', fontsize=12)
plt.show()
```



Review Length vs Sentiment

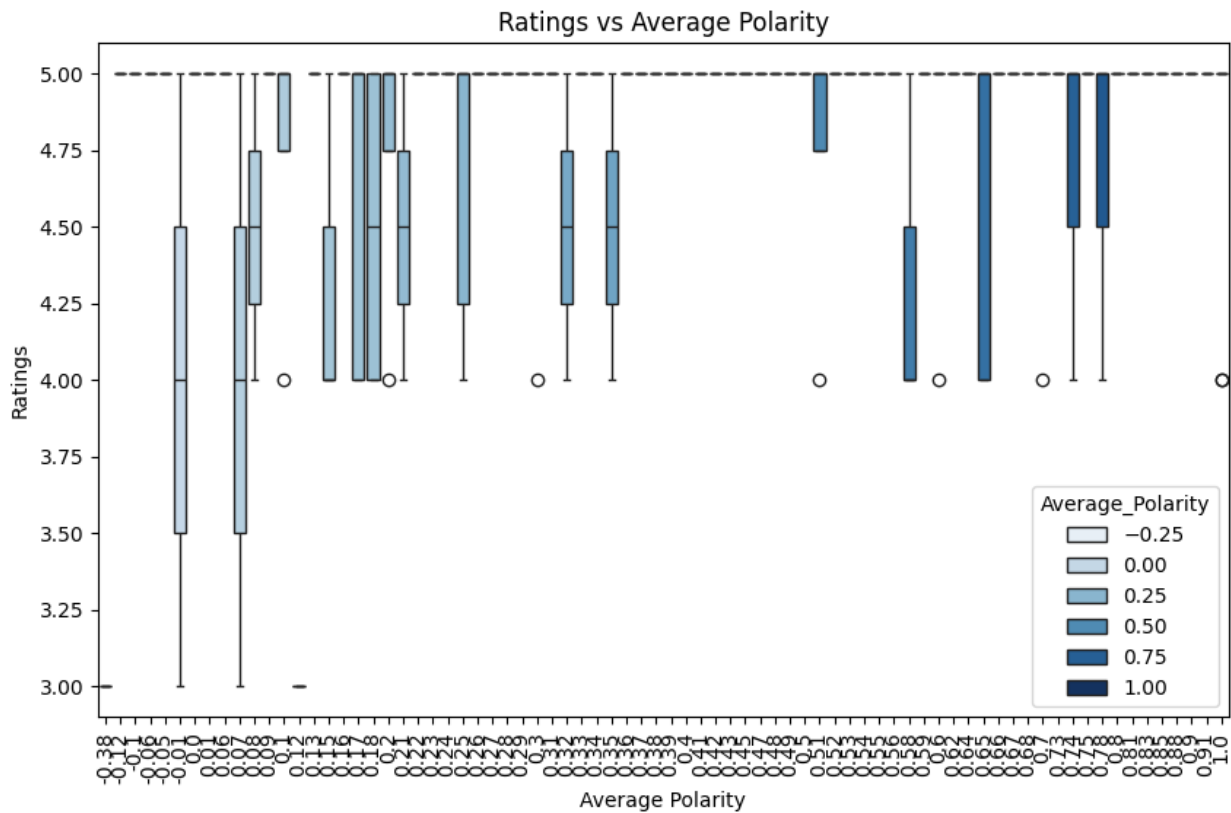
- Higher-rated reviews tend to be more detailed, with users elaborating on their experiences.
- Neutral reviews are spread across different ratings, suggesting that star ratings don't always align with textual sentiment.
- Negative feedback is usually brief, but variations exist due to individual opinions and expectations.

```
# Plotting ratings vs average polarity
plt.figure(figsize=(10, 6))
sns.boxplot(x='Average_Polarity', y='Ratings', data = df1, hue =
```

```

'Average_Polarity', palette='Blues')
plt.title('Ratings vs Average Polarity')
plt.xlabel('Average Polarity')
plt.ylabel('Ratings')
plt.xticks(rotation=90)
plt.show()

```



Ratings vs Average Polarity

- Positive sentiment reviews show the widest range in review length, with a few detailed outliers.
- Extremely positive feedback is generally concise and to the point.
- Neutral reviews have a narrower distribution, similar to extremely positive ones.
- Negative reviews are shorter on average, but their length varies more than neutral ones.
- Overall, longer reviews tend to have more positive sentiment, while negative ones are typically brief but impactful.

5. Reporting

Summarize findings with:

- **Data collection & cleaning overview**
- **Sentiment distribution & trends**
- **Key insights & product issues**

- Recommendations for improvement & marketing

Sentiment Analysis Report

1. Data Collection & Cleaning

Source:

-- Flipkart customer reviews were extracted using Selenium and BeautifulSoup.

Processing:

-- Text was cleaned by removing unnecessary characters and standardizing formatting.

-- Tokenization was applied for further analysis.

-- Sentiments were categorized as **positive, neutral, or negative**.

2. Sentiment Findings

Review Sentiment Distribution:

-- Most reviews are **positive**, with a small percentage being **extremely positive**.

-- **Neutral and negative feedback** is minimal in comparison.

Ratings & Sentiment:

-- **Higher ratings** generally reflect **positive sentiment**.

-- **Lower ratings** often indicate **neutral or negative experiences**, signaling areas of dissatisfaction.

3. Key Insights

What customers love:

-- **Design, camera quality, and performance** stand out as strong positives.

-- **Battery life improvements** are frequently praised.

Common concerns:

-- **Pricing complaints** and occasional **packaging/delivery issues**.

-- Some users report **compatibility concerns** with accessories and **minor software glitches**.

4. Recommendations

Product Enhancements:

-- Address **minor software issues** to improve user satisfaction.

-- Investigate **accessory compatibility concerns** to ensure seamless usability.

Marketing Strategies:

-- Highlight **camera quality, battery performance, and sleek design** in promotions.

-- Offer **EMI, exchange deals, or limited-time discounts** to make pricing more attractive.

Operational Improvements:

- Focus on **improving delivery services** to reduce packaging complaints.
- Maintain **proactive customer feedback monitoring** to address new issues quickly

This ensures that Flipkart can refine its offerings and improve the customer experience for future buyers.